

Amendments to the Specification:

Please replace the paragraph beginning at page 4, line 12, with the following amended paragraph:

Figure 1 illustrates a system 100 in accordance with an embodiment of the present invention. The system 100 includes a plurality of hardware components 10-12, a processor or central processing unit (CPU) 20, a BIOS (Basic Input Output System) 30, a timer 40, and a bus 50. Examples of hardware components are a card, chipset, register, hard disk, keyboard, monitor, communication port, RAM (random access memory) memory, mouse, floppy drive, etc. A hardware component can also have a BIOS (e.g., BIOS 11A). The system 100 can be implemented in several different configurations. Examples of these configurations include a computer system, a workstation, a DVD (Digital Video Disc) player, a game console, a graphics chipset, and a graphics card.

Please replace the paragraph beginning at page 6, line 22, with the following amended paragraph:

If a BIOS initialization task, which is currently executing, requests a waiting period (e.g., 10 milliseconds, 50 milliseconds, etc.) by making a call to the BIOS kernel 35, the BIOS initialization task is suspended and is designated a waiting task for the duration of the waiting period. The BIOS kernel 35 can set the timer 40 based on the requested waiting period. If the waiting period has elapsed for several waiting tasks at the same time, the waiting task having the highest priority value (and is also greater than the priority value of the executing task) is started again and designated as an executing task, as shown by arrow

220. If the waiting period has elapsed for any waiting task having a lower priority value than the priority value of the executing task, the waiting task remains suspended and is designated as an interrupted task, as shown by arrow 225.

Please replace the paragraph beginning at page 8, line 4, with the following amended paragraph:

Referring to Figure 3A, the plurality of BIOS initialization tasks includes Task1 and Task2. These tasks are designated as unstarted (U) 301 and 311 tasks at time=T1. However, at time=T2, Task1 is selected, started, and designated as an executing (E) 302 task because Task1 has the highest priority value of the unstarted (U) 301 and 311 tasks, Task1 does not have any dependent initialization tasks, and there are no interrupted tasks. At time=T3, Task1 requests a waiting period by making a call to the BIOS kernel 35. Thus, Task1 is suspended and is designated as a waiting (W) 303 task. Moreover, arrow C1 indicates that the BIOS kernel 35 makes a cooperative multitasking transition to Task2 at time=T3. That is, Task2 is started and designated as an executing (E) 312 task.

Please replace the paragraph beginning at page 8, line 14, with the following amended paragraph:

At time=T4, the waiting period of Task1 has elapsed. Hence, arrow P1 indicates that the BIOS kernel 35 makes a preemptive multitasking transition to

Task1 at time= T_4 . That is, Task1 is started again and designated as an executing (E) 304 task since Task1 has a higher priority value than Task2. Moreover, since the waiting period has elapsed for a waiting (W) 303 task (Task1) having a higher priority value than the priority value of the executing (E) 312 task (Task2), Task2 is suspended and designated as an interrupted (I) 313 task.

Please replace the paragraph beginning at page 8, line 22, with the following amended paragraph:

At time= T_5 , Task1 requests a waiting period by making a call to the BIOS kernel 35. Thus, Task1 is suspended and is designated as a waiting (W) 305 task. Moreover, arrow C2 indicates that the BIOS kernel 35 makes a cooperative multitasking transition to Task2 at time= T_5 . That is, Task2 is started again and designated as an executing (E) 314 task since the interrupted (I) 313 task (Task2) has the highest priority value of the interrupted tasks and there is no executing task.

Please replace the paragraph beginning at page 9, line 6, with the following amended paragraph:

Referring to Figure 3B, the plurality of BIOS initialization tasks includes Task1, Task2, and Task3. At time= T_1 , Task1 and Task2 are designated as waiting (W) 321 and 331 tasks while Task3 is designated as an executing (E) 341 task. There are no unstarted tasks. At time= T_2 , the waiting period of Task2

has elapsed. Hence, arrow P1 indicates that the BIOS kernel 35 makes a preemptive multitasking transition to Task2 at time=T2. That is, Task2 is started again and designated as an executing (E) 332 task since Task2 has a higher priority value than Task3. Moreover, since the waiting period has elapsed for a waiting (W) 331 task (Task2) having a higher priority value than the priority value of the executing (E) 341 task (Task3), Task3 is suspended and designated as an interrupted (I) 342 task.

Please replace the paragraph beginning at page 9, line 16, with the following amended paragraph:

At time=T3, the waiting period of Task1 has elapsed. Hence, arrow P2 indicates that the BIOS kernel 35 makes a preemptive multitasking transition to Task1 at time=T3. That is, Task1 is started again and designated as an executing (E) 322 task since Task1 has a higher priority value than Task2. Moreover, since the waiting period has elapsed for a waiting (W) 321 task (Task1) having a higher priority value than the priority value of the executing (E) 332 task (Task2), Task2 is suspended and designated as an interrupted (I) 333 task.

Please replace the paragraph beginning at page 10, line 1, with the following amended paragraph:

At time=T4, Task1 requests a waiting period by making a call to the BIOS kernel 35. Thus, Task1 is suspended and is designated as a waiting (W) 323

task. Moreover, arrow C1 indicates that the BIOS kernel 35 makes a cooperative multitasking transition to Task2 at time=T4. That is, Task2 is started again and designated as an executing (E) 334 task since the interrupted (I) 333 task (Task2) has the highest priority value of the interrupted (I) 333 and 342 tasks (Task2 and Task3) and there is no executing task.

Please replace the paragraph beginning at page 10, line 8, with the following amended paragraph:

Referring to Figure 3C, the plurality of BIOS initialization tasks includes Task1, Task2, and Task3. At time=T1, Task1, Task2, and Task3 are designated as waiting (W) 351, 361, and 371 tasks. At time=T2, the waiting period has elapsed for several waiting (W) 351 and 371 tasks (Task1 and Task3) at the same time, the waiting (W) 351 task (Task1) having the highest priority value is started again and designated as an executing task (E) 352 while Task3 remains suspended and is designated as an interrupted (I) 372 task. Moreover, at time=T3, the waiting period has elapsed for Task2 which has a lower priority value than the priority value of the executing (E) 352 task (Task1), the Task2 remains suspended and is designated as an interrupted (I) 362 task.

Please replace the paragraph beginning at page 10, line 18, with the following amended paragraph:

Referring to Figure 3D, the plurality of BIOS initialization tasks includes Task1, Task2, and Task3. At time=T1, Task1 is designated as executing (E) 381

task while Task2 and Task3 are designated as unstarted (U) 391 and 394 tasks. Moreover, Task2 starts only if Task1 has completed.

Please replace the paragraph beginning at page 11, line 1, with the following amended paragraph:

At time= T_2 , Task1 requests a waiting period by making a call to the BIOS kernel 35. Thus, Task1 is suspended and is designated as a waiting (W) 3382 task. Moreover, arrow C1 indicates that the BIOS kernel 35 makes a cooperative multitasking transition to Task3 at time= T_2 . That is, Task3 is started and designated as an executing (E) 395 task because Task3 has the highest priority value of the unstarted tasks (U) 391 and 394 without any dependent initialization tasks that have not been completed and there are no interrupted tasks.

Please replace the paragraph beginning at page 11, line 9, with the following amended paragraph:

At time= T_3 , the waiting period of Task1 has elapsed. Hence, arrow P1 indicates that the BIOS kernel 35 makes a preemptive multitasking transition to Task1 at time= T_3 . That is, Task1 is started again and designated as an executing (E) 383 task since Task1 has a higher priority value than Task3. Moreover, since the waiting period has elapsed for a waiting (W) 382 task (Task1) having a higher priority value than the priority value of the executing (E)

395 task (Task3), Task3 is suspended and designated as an interrupted (I) 396 task.

Please replace the paragraph beginning at page 11, line 17, with the following amended paragraph:

At time=T4, Task1 has completed and is designated as a completed (C) 384 task. Moreover, arrow C2 indicates that the BIOS kernel 35 makes a cooperative multitasking transition to Task2 at time=T4. That is, Task2 is started and designated as an executing (E) 392 task because Task2 has the highest priority value of the unstarted (U) 391 tasks, the dependent initialization task (Task1) has been completed, and Task2 has a higher priority value than the interrupted (I) 396 task (Task3).

Please replace the paragraph beginning at page 12, line 1, with the following amended paragraph:

Figure 4 illustrates a flow chart showing a method 400 for multitasking BIOS initialization tasks in accordance with an embodiment of the present invention. In an embodiment, the present invention is implemented as computer-executable instructions for performing this method 400. The computer-executable instructions can be stored in any type of computer-readable medium, such as a magnetic disk, CD-ROM (compact disc - read only memory), an optical medium, a floppy disk, a flexible disk, a hard disk, a magnetic tape, a RAM (random access memory), a ROM (read only memory), a PROM (programmable

read only memory), an EPROM (erasable programmable read only memory), a flash-EPROM (flash - erasable programmable read only memory), or any other medium from which a computer can read. It should be understood that the method 400 is applicable to tasks other than BIOS initialization tasks.

Please replace the paragraph beginning at page 13, line 6, with the following amended paragraph:

If the executing task has not requested the waiting period, after Step 430 the method 400 proceeds to Step 440. At Step 440, it is determined whether a waiting period for a waiting task has elapsed. If a waiting period for a waiting task has elapsed, the method 400 proceeds to Step 460. Otherwise, the method 400 proceeds to Step 450.